View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Procedia Computer Science

Volume 29, 2014, Pages 1345–1355

ICCS 2014. 14th International Conference on Computational Science

# Using Hyperheuristics to Improve the Determination of the Kinetic Constants of a Chemical Reaction in Heterogeneous Phase

José-Matías Cutillas-Lozano and Domingo Giménez

Departamento de Informática y Sistemas, University of Murcia, Spain
{josematias.cutillas,domingo}@um.es

**Abstract**

The reaction in the human stomach when neutralizing acid with an antacid tablet is simulated and the evolution over time of the concentration of all chemical species present in the reaction medium is obtained. The values of the kinetic parameters of the chemical reaction can be determined by integrating the equation of the reaction rate. This is a classical optimization problem that can be approached with metaheuristic methods. The use of a parallel, parameterized scheme for metaheuristics facilitates the development of metaheuristics and their application. The unified scheme can also be used to implement hyperheuristics on top of parameterized metaheuristics, so selecting appropriate values for the metaheuristic parameters, and consequently the metaheuristic itself. The hyperheuristic approach provides satisfactory values for the metaheuristic parameters and, consequently, satisfactory metaheuristics for the problem of determining the kinetic constants.

*Keywords:* Hyperheuristics, Kinetic constants determination, Chemical reaction

## 1 Introduction

In a previous work [3] we estimated the kinetic parameters of a chemical reaction with metaheuristic methods. Now, as a continuation, we focus on the use of hyperheuristics, which are automatic search algorithms for identifying good metaheuristics [2, 10]. A chemical reaction in heterogeneous phase that takes place in a batch stirred tank reactor was studied [7], and the processes occurring in the human stomach when neutralizing the acid with an antacid tablet was simulated. It is a reaction combined with mass transfer of carbonate ions present in the solid phase upon contact with an acid solution. The solid phase consists of an antacid tablet, which contains a certain amount of calcium carbonate. The reaction medium represents the approximate pH of the human stomach (with values between 3 and 4). The kinetic parameters of the reaction can be determined using the Euler numerical method.

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

© The Authors. Published by Elsevier B.V.
doi:10.1016/j.procs.2014.05.121

This is a classical optimization problem consisting of the estimation of the kinetic parameters that best fit the experimental data for a given reaction mechanism. We can find several ways of addressing the problem in the literature. Many of them use statistical methods based on linear and non-linear regression analysis [8] and numerical solution procedures for differential equations based on the Gauss-Newton method [9]. There is a wide range of mathematical software which implements these methods in computation modules. In fact, in a previous work [3] the "Excel Solver" method was used to obtain preliminary results including the approximate ranges of some parameters. More recently, metaheuristic methods have begun to be used to address these problems [5, 6]. As far as we know, there is no evidence of hyperheuristics being used to determine parameters of chemical reactions. The use of a unified parameterized scheme for metaheuristics facilitated the development of metaheuristics and their application by reusing the basic functions, and the parallelism accelerated the solution in large simulations. The scheme has been applied successfully to other problems [1, 4].

The same unified scheme can also be used to implement hyperheuristics on top of parameterized metaheuristics, selecting appropriate values for the metaheuristic parameters, and consequently the metaheuristic itself. A criterion based on the fitness is used to automatically select good metaheuristics for each problem from certain training instances. The execution time is also considered when these metaheuristics are applied to other sets of test instances. The results with the application of the hyperheuristic approach introduced here can be compared with earlier results in which the metaheuristics are tuned with the parameterized scheme.

The rest of the paper is organized as follows. Section 2 summarizes the problem to be solved and describes the terminology used. Section 3 shows the general parameterized scheme of metaheuristics and hyperheuristics and its application to the problem in hand. The experimental results are shown in Section 4. Section 5 summarizes the conclusions and proposes future work lines.

## 2 A Problem of Determination of the Kinetic Constants of a Chemical Reaction in Heterogeneous Phase

Here we briefly summarize the problem of determining the kinetic constants detailed in [3]. When chemical reactions occur in heterogeneous phase the variables that affect the reaction rate are not only temperature, pressure or composition; mass transfer rate is also important and should be included in the terms of the reaction rate. When reaction rates in complex systems are compared or combined, we have to take into account that if the change of property is effected by several mutually independent parallel paths, the overall rate is simply the sum of all individual rates. The kinetic of dissolution of calcium carbonate is a function of the concentration of the various carbonate species in the solution and, therefore, a function of the partial pressure of carbon dioxide and pH. A model for the dissolution of calcium carbonate was developed in [11]. Depending on the value of the pH, there are four different ways in which the dissolution of calcium carbonate occurs: by reaction with acetic acid, by reaction with carbonic acid, and by the hydrolysis reaction (the fourth path may be neglected in the pH range considered). Therefore the variation of moles of calcium over time is:

$$\frac{1}{V}\frac{dN_{Ca^{2+}}}{dt} = -k_1 a^{n_1}\left[H_3O^+\right]^{n_2} - k_2 a^{n_3}\left[H_2CO_3\right]^{n_4} - k_3 \quad (1)$$

where $k_1$, $k_2$ and $k_3$ are the combined reaction rate constants, $n_1$, $n_2$, $n_3$ and $n_4$ are the reaction orders, and $a$ is the area of the tablet, which is known for each increase of time. The integration

Calculate Fitness($k_1$, $k_2$, $k_3$, $n_1$, $n_2$, $n_3$, $n_4$):
**for** $i = 0 \rightarrow N$ **do**
Calculate at instant $i$:
$[Ca^{2+}], a, [H_3O^+], [HCO^-], [H_2CO_3], pH_{cal}, \Delta[Ca^{2+}], [CH_3COOH], [CH_3COO^-]$
$Fitness = Fitness + (pH_{exp,i} - pH_{cal,i})^2$
**end for**

**Algorithm 1:** Calculation of the fitness function for an individual

```
Initialize(S,ParamIni)
while (not EndCondition(S,ParamEnd)) do
  SS=Select(S,ParamSel)
  SS1=Combine(SS,ParamCom)
  SS2=Improve(SS1,ParamImp)
  S=Include(SS2,ParamInc)
end while
```

**Algorithm 2:** Parameterized metaheuristic scheme

of the equation for a certain value of time, $t_j$, gives the increase of calcium in the solution as a function of the concentrations of protons and carbonic acid.

The values of the parameters can be approached with metaheuristics. An individual is represented by a real vector of size seven. The ranges of values for the constants are set following empirical criteria. Every time we have to evaluate the fitness of an individual, we must solve the whole chemical system. The calculation of the fitness function is presented in Algorithm 1, where we have a loop of $N$ time intervals that calculates the value of each variable of the chemical problem for next time instant $i + 1$ depending on the value of the variable in the previous instant $i$.

# 3 Parameterized Schemes for Metaheuristics and Hyperheuristics

The use of a unified parameterized scheme for metaheuristics facilitates the development of metaheuristics and their application [1]. The selection of the appropriate values of parameters can be made through a hyperheuristic method also developed with the parameterized metaheuristic scheme. In this section, a description of the functions and parameters of the metaheuristic scheme is presented, as is its use for hyperheuristic implementations.

## 3.1 A Parameterized Scheme for Metaheuristics

A parameterized metaheuristic scheme was developed in a previous work [1]. The scheme (Algorithm 2) considers a set of basic functions whose instantiation, by selecting the appropriate values of the *Metaheuristic parameters* ($ParamX$), determines the particular metaheuristic. The selection of these values can be made through a hyperheuristic method also developed with the parameterized metaheuristic scheme. The arguments `S`, `SS`, `SS1`, and `SS2` correspond to the sets of solutions that the method manipulates in successive iterations. The same scheme represents several metaheuristics and allows reuse of the functions and variables.

The meaning and number of the parameters depend on the basic metaheuristics considered

and on the implementation of the basic functions. The basic metaheuristics used in our work are GRASP, Genetic Algorithms (GA), Scatter Search (SS) and Tabu Search (TS). The basic functions and the meaning of the parameters in the parallel parameterized scheme are briefly commented on:

- `Initialize:` Valid random elements (see section 2) are generated to form an initial set with $INEIni$ elements. A smaller subset with $FNEIni$ elements is selected for the iterations in Algorithm 2. In some metaheuristics (for example, SS and GRASP) some of the initial elements are improved by using, for example, a local search or a greedy approach. A parameter $PEIIni$ indicates the percentage of elements to be improved, and the improvement may be more or less intense, which is represented by an intensification parameter, $IIEIni$. The parameter $STMIni$ is used for the extension of Tabu short-term memory in the initialization improvement. So, we consider five parameters in the initialization function: $ParamIni = \{INEIni, FNEIni, PEIIni, IIEIni, STMIni\}$.

- `EndCondition:` The end condition is common to the different metaheuristics; it consists of a maximum number of iterations ($MNIEnd$) or a maximum number of iterations without improving the best solution ($NIREnd$), and $ParamEnd = \{MNIEnd, NIREnd\}$.

- `Select:` The elements can be grouped into two sets, the best and worst according to the objective function. The number of best elements will be $NBESel$ and that of worst elements $NWESel$, and normally $NBESel + NWESel = FNEIni$. So, $ParamSel = \{NBESel, NWESel\}$.

- `Combine:` The total number of elements to obtain by combination is $2(NBBCom+NBWCom+NWWCom)$, where the three parameters represent the number of combinations of the best with the best elements, the number of the best with the worst and the number of the worst with the worst. They constitute the set of parameters for the function: $ParamCom = \{NBBCom, NBWCom, NWWCom\}$.

- `Improve:` As in the improvement in the initialization, $PEIImp$, $IIEImp$ and $SMIImp$ represent the percentage of elements to be improved, the intensification of the improvement and the short-term memory in the improvement of the elements generated in the combination; and $PEDImp$, $IDEImp$ and $SMDImp$ represent the corresponding values in a diversification, which is equivalent to the mutation in the GA. So, $ParamImp = \{PEIImp, IIEImp, SMIImp, PEDImp, IDEImp, SMDImp\}$.

- `Include:` The $NBEInc$ best elements are maintained in the reference set, and the other $FNEIni - NBEInc$ to be included are selected from the remaining elements. $LTMInc$ is a Tabu parameter (long-term memory) that allows the tracking of individuals most frequently explored. So, $ParamInc = \{NBEInc, LTMInc\}$.

The set of metaheuristic parameters is the union of the six sets of parameters of the basic functions in the scheme. There is a set with 20 metaheuristic parameters with which it is possible to experiment to hybridize, mix and adapt the metaheuristics to the target problem.

## 3.2 Hyperheuristics Based on Parameterized Metaheuristic Schemes

The same parameterized scheme of metaheuristics is used here for the implementation of hyperheuristics, which will work by searching the metaheuristic space determined by the values

of the metaheuristic parameters, and which can in turn use the same metaheuristic scheme for hyperheuristics development.

In the hyperheuristic an individual or element is represented by an integer vector $MetaheurParam$ that encodes the set of parameters that characterizes a metaheuristic using the scheme in Algorithm 2. The number and meaning of the parameters is determined by the basic metaheuristics integrated in the scheme, the implementation of the basic functions and the parameters that can be varied in the functions. Along with those considered in the previous subsection, we have a total of 20 metaheuristic parameters, so one element in the hyperheuristic will have 20 components, with values in ranges which depend on the possible values of the parameter itself and also on some restrictions imposed on the hyperheuristic to reduce the search space. For example, there is no limit in the initial population size in the metaheuristic ($INEIni$), but an upper limit should be imposed, and this limit can be small so as to reduce the hyperheuristic execution time. The elements and other aspects of the hyperheuristics will be explained in the following sections with the application to the optimization problem.

The set of individuals constitutes the reference set, which means a set of metaheuristics, with each metaheuristic being the combination/hybridation of basic metaheuristics given by the values in $MetaheurParam$. The fitness value in the hyperheuristic for an element $MetaheurParam$ is based on that obtained when the metaheuristic with the parameters in $MetaheurParam$ is applied to the problem for which the metaheuristic is being obtained. As in the case of metaheuristics, the set of hyperheuristic parameters is the union of the six sets of parameters of the basic functions in the scheme. Thus, we have a set of 20 parameters with which we can experiment and apply the hyperheuristics to select good metaheuristics for the target problems.

Our aim is to minimize the fitness function and so obtain the combination of the metaheuristic parameters which gives the best fitness function for different instances of the problem in question. There are different possibilities for the fitness function, but one based on several instances of the problem executed together was chosen. Thus, FitSP1E (Fitness with several problem inputs in one execution) is a way of reducing the dependence on the input and the increase of the execution time by applying the metaheuristics to only one input, and changing the particular input during the hyperheuristic application. The same problem is used in each iteration, but the problem to which the metaheuristics are applied changes in each iteration or group of iterations.

The characteristics of the basic functions of the algorithm for hyperheuristics are commented on below:

- `Initialize:` Due to the high computational cost of hyperheuristics, the number of elements in the initial set and in the reference set is normally smaller than for metaheuristics. So, the application of local search methods or population based methods with small populations may be preferable. Valid random elements are generated, with different ranges for the different parameters, and with shorter ranges than for metaheuristics.

- `EndCondition:` The end condition may be common to that of metaheuristics, but the limits on the numbers of iterations will be lower.

- `Select:` The selection of elements is made as for metaheuristics. The only difference is the fitness function used to classify the elements.

- `Combine:` As for metaheuristics, pairs of best, worst and best-worst elements can be combined, but the combination is normally problem-dependent, so satisfactory combi-

| | INEIni | FNEIni | PEIIni | IIEIni | STMIni | NBESel | NWESel | NBBCom | NBWCom |
|---|---|---|---|---|---|---|---|---|---|
| Hre | 5 | 5 | 50 | 3 | 2 | 3 | 2 | 2 | 3 |
| Hge | 20 | 20 | 0 | 0 | 0 | 20 | 0 | 10 | 0 |

| | NWWCom | PEIImp | IIEImp | SMIImp | PEDImp | IDEImp | SMDImp | NBEInc | LTMInc |
|---|---|---|---|---|---|---|---|---|---|
| Hre | 2 | 50 | 3 | 2 | 10 | 5 | 2 | 3 | 5 |
| Hge | 0 | 0 | 0 | 0 | 10 | 5 | 0 | 20 | 0 |

Table 1: Values of the hyperheuristic parameters used for the selection of metaheuristics: Reduced Hybrid Hyperheuristic (Hre) and Genetic Algorithm based Hyperheuristic (Hge).

nations for the metaheuristic selection problem should be obtained. There are various possibilities and a combination by groups was chosen: the parameters are organized and selected in groups. For example, $GroupIni = \{INEIni, FNEIni, STMIni\}$, $GroupImpIni = \{PEIIni, IIEIni\}$, etc, and groups of parameters are selected from the ascendants to generate the descendants.

- `Improve:` This works in the same way as for metaheuristics, but with lower values of the parameters to reduce the execution time.
- `Include:` The only differences with metaheuristics are the fitness function for selection of the best elements, and a lower value of the parameter for the Tabu long-term memory.

As in the case of metaheuristics, the set of hyperheuristic parameters is the union of the six sets of parameters of the basic functions in the scheme. Thus, we have a set of 20 parameters with which we can experiment and apply the hyperheuristics to select good metaheuristics for the target problems.

# 4 Experimental Results

In a previous work [3], we estimated the kinetic parameters of a chemical reaction with metaheuristic methods. Our work focuses on analyzing the applicability of hyperheuristics and their suitability as a general optimization tool, independent of the problem, in terms of fitness for the problem. We are interested in minimizing the fitness function but, if possible, without increasing the execution time much, so the inverse of the product of the fitness and the execution time is used as a common indicator (CI). Therefore, high values are desirable for this indicator.

Since the hyperheuristics are developed using the parameterized scheme of metaheuristics, it suffices to fix the metaheuristic parameters of the hyperheuristic (we call them hyperheuristic parameters) to specific values. Table 1 shows the two types of hyperheuristics that will serve as references for selecting good metaheuristics for each problem:

- We have chosen a hyperheuristic with a genetic algorithm approach (Hge) which works with a small population and improvement is made only in the diversification, which means the elements obtained with mutations are improved to reduce the possibility of their rapid death. The values of the Tabu parameters are set to zero.
- A reduced hybrid hyperheuristic (Hre) is also considered. It is a hybrid scheme with low values of the metaheuristic parameters to reduce the execution time.

| | INEIni | FNEIni | PEIIni | IIEIni | STMIni | NBESel | NWESel | NBBCom | NBWCom |
|---|---|---|---|---|---|---|---|---|---|
| GR | 200 | 1 | 100 | 50 | 0 | 0 | 0 | 0 | 0 |
| GA | 100 | 100 | 0 | 0 | 0 | 100 | 0 | 50 | |
| SS | 100 | 20 | 100 | 50 | 0 | 10 | 10 | 90 | 100 |
| TS | 200 | 1 | 100 | 10 | 5 | 1 | 0 | 0 | 0 |
| Mhy | 50 | 15 | 100 | 15 | 2 | 8 | 7 | 15 | 20 |

| | NWWCom | PEIImp | IIEImp | SMIImp | PEDImp | IDEImp | SMDImp | NBEInc | LTMInc |
|---|---|---|---|---|---|---|---|---|---|
| GR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| GA | 0 | 0 | 0 | 0 | 10 | 5 | 0 | 100 | 0 |
| SS | 90 | 100 | 5 | 0 | 0 | 0 | 0 | 10 | 0 |
| TS | 0 | 100 | 5 | 5 | 0 | 0 | 0 | 1 | 20 |
| Mhy | 15 | 100 | 5 | 2 | 20 | 5 | 2 | 8 | 3 |

Table 2: Values of the parameters for the four pure metaheuristics and the hybrid metaheuristic (Mhy) considered.

| | INEIni | FNEIni | PEIIni | IIEIni | STMIni | NBESel | NWESel | NBBCom | NBWCom |
|---|---|---|---|---|---|---|---|---|---|
| Lower | 5 | 5 | 0 | 1 | 0 | 2 | 2 | 5 | 5 |
| Upper | 200 | 100 | 100 | 20 | 15 | 100 | 100 | 100 | 100 |

| | NWWCom | PEIImp | IIEImp | SMIImp | PEDImp | IDEImp | SMDImp | NBEInc | LTMInc |
|---|---|---|---|---|---|---|---|---|---|
| Lower | 5 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 0 |
| Upper | 100 | 100 | 20 | 15 | 100 | 10 | 15 | 100 | 15 |

Table 3: Lower and upper limits of the metaheuristic parameters for the selection of individuals by the hyperheuristics.

To verify the effectiveness of the hyperheuristics, it seems reasonable to compare the fitness obtained with the metaheuristic selected by the hyperheuristics with those obtained with some pure metaheuristics applied directly to the optimization problem. Instances of the parameters corresponding to the four basic metaheuristics considered have been chosen. In a previous work [3] we have experimented in depth with hybrid metaheuristics to obtain the most efficient ones for our optimization problem. Therefore, it also seems appropriate to compare the best results obtained with the best hybrid metaheuristic with those obtained with the hyperheuristic.

Table 2 shows the values selected for the parameters for the four basic metaheuristics and the best hybrid metaheuristics obtained in a previous work for the optimization problem (Mhy) as a combination of the four pure metaheuristics. The values of the metaheuristic parameters are in general higher than the corresponding values for the hyperheuristics, because the application of one metaheuristic requires much lower execution times than that of one hyperheuristic.

In principle, any values of the parameters can determine a metaheuristic, but it is preferable to establish ranges that allow us to compare the results obtained with the application of the hyperheuristic with those obtained with pure metaheuristics. Therefore, the hyperheuristic will select metaheuristic parameters for each function of the scheme in the ranges of values presented in table 3, where we consider that lower values greater or equal to zero are preferable for some parameters in most cases. The two EndCondition parameters were fixed to 10.

| | INEIni | FNEIni | PEIIni | IIEIni | STMIni | NBESel | NWESel | NBBCom | NBWCom |
|---|---|---|---|---|---|---|---|---|---|
| $Hre_{FitSP1E}$ | 24 | 17 | 45 | 2 | 12 | 6 | 9 | 99 | 1 |
| $Hge_{FitSP1E}$ | 145 | 32 | 66 | 11 | 8 | 5 | 21 | 71 | 77 |

| | SMIImp | PEDImp | IDEImp | SMDImp | NBEInc | LTMInc | NWWCom | PEIImp | IIEImp |
|---|---|---|---|---|---|---|---|---|---|
| $Hre_{FitSP1E}$ | 8 | 86 | 10 | 11 | 86 | 4 | 9 | 16 | 9 |
| $Hge_{FitSP1E}$ | 62 | 72 | 14 | 8 | 74 | 8 | 12 | 27 | 3 |

Table 4: Values of the metaheuristic parameters obtained when applying the hybrid-reduced (Hre) and the genetic hyperheuristics (Hge) to three training instances of the problem varying the instance for each iteration in the same execution (FitSP1E).

In section 2 we summarized the problem of determination of the kinetic constants of a chemical reaction. We continue the validation of the method with the application of hyperheuristics to this problem. In the previous study, six different instances of the problem were considered (S1 to S6). Now we continue to consider these, and more details about the specific characteristics of each one can be found in [3].

Table 4 shows the metaheuristic parameters obtained when different settings of the hyperheuristics are applied to three training instances (S1, S2 and S3) of the optimization problem using FitSP1E as fitness calculation criterion. We observe a significant difference in the number of elements constituting the initial set ($INEIni$), although the number of elements selected from the reference set ($FNEIni$) is similar and low. Almost 50% of the elements are improved in the initialization ($PEIIni$), with small improvement intensities ($IIEIni$) and intermediate values of tabu short-term memory ($STMIni$). Notably, the genetic hyperheuristic has some combination parameter values ($NBWCom$ and $NWWCom$) that are significantly higher than those of the reduced hyperheuristic. The improvement ($PEIImp$, $IIEImp$) and diversification ($PEDImp$, $IDEImp$) are important; intermediate tabu memory values ($SMIImp$, $SMDImp$ and $LTMInc$) give satisfactory results; and there is priority in the selection of the best elements for the reference set ($NBEInc$).

Figure 1 shows the results of the application of the hyperheuristic to three instances of the problem used for the tests (S4, S5 and S6 series). The best metaheuristics were obtained from the validation set (S1, S2 and S3) in one execution of each hyperheuristic. On average, metaheuristics obtained with FitSP1E appreciably improve the results achieved by applying the pure metaheuristics, and they provide fitness values that are close to the best obtained by applying the hyperheuristic to the validation problems directly.

The results obtained with the two hyperheuristic configurations with the FitSP1E methodology are compared in table 5 (a). The table shows the mean of the fitness, sequential time and common indicator obtained by applying the technique to the three test problems S4, S5 and S6. The best indicator (CI) comes from the Hre because it has the lowest execution time, with similar fitness values.

We can compare the hyperheuristic with previous results to see whether it improves fitness. The last column of table 5 (b) gives the results obtained by applying the best (hybrid) metaheuristic (Mhy in table 2) we obtained in previous works to three different problem sizes. We can see that in all cases the two hyperheuristic configurations improve these results. This corroborates the need for a hyperheuristic as a good automatic search algorithm both for individual problems and to avoid problem dependency.

Finally, in figure 2, a statistical summary of the means obtained by applying different

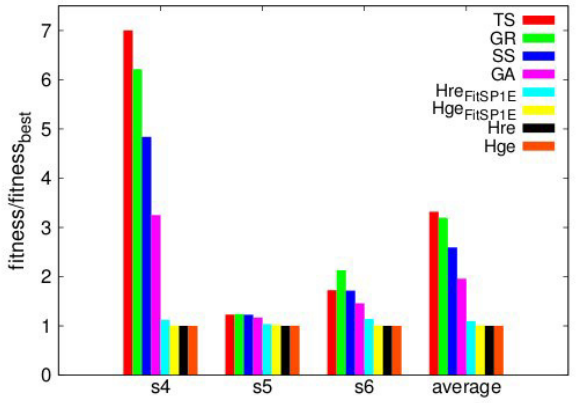



Figure 1: Quotient of fitness with respect to the best fitness, for various instances of the problem, for four pure metaheuristics, with the optimum hybrid metaheuristics selected by the hyperheuristics for variable problems in one execution (FitSP1E), and with the direct application of the hyperheuristics to each problem.

| | | Hre | Hge |
|---|---|---|---|
| FitSP1E | (f) | 2.1444 | 2.0348 |
| | (t) | 59.93 | 783.17 |
| | (CI) | 77.81 | 6.28 |

(a) Indicators for hyperheuristics

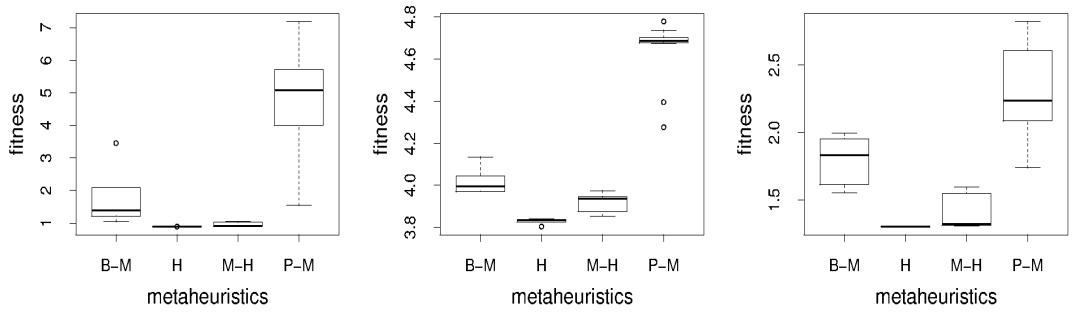
| | Hre | Hge | Mhy |
|---|---|---|---|
| S4 | 0.8937 | 0.8931 | 1.7600 |
| S5 | 3.8209 | 3.8311 | 4.0178 |
| S6 | 1.3023 | 1.3022 | 1.7953 |
| Mean | 2.0055 | 2.0088 | 2.5243 |

(b) Hyperheuristics - Best metaheuristic

Table 5: Comparison of (a) the mean fitness (f), time (t) and common indicator ($CI = 10^4/(f \cdot t)$) obtained when applying the best metaheuristics reached with FitSP1E for two hyperheuristic configurations to the instances S4, S5 and S6 of the problem, and (b) the fitness obtanined when executing two hyperheuristic configurations directly to the three test problems with those obtained when applying the best hybrid metaheuristic (Mhy).

metaheuristic algorithms to several sizes of the problem is presented. The results are grouped into various sets for each size with a variable number of points on each set (between four and twelve): the direct application of the two hyperheuristic configurations (H), the set of metaheuristics obtained from hyperheuristics (M-H) and applied with fitness FitSP1E, the application of instances of the best metaheuristic (B-M) not selected automatically (combination Mhy in table 2), and the set of the four pure metaheuristics (P-M). A relatively small number of points was considered in some cases because of the high execution time in some algorithms with very small variances. The Kruskal-Wallis test revealed statistical differences in the means for the sets of instances in the three problem sizes considered.

The Wilcoxon rank sum test with continuity correction was applied at a significance level $\alpha = 0.05$. The best algorithm in all problem sizes was the hyperheuristic (H) applied directly to each instance, with the other algorithms being worse in all cases. Furthermore, there are significant differences between all pairs of algorithms applied in each problem size. This confirms




(a) PKICO S4 (b) PKICO S5 (c) PKICO S6

Figure 2: Statistical summary of the fitness means obtained by applying different metaheuristic algorithms to several sizes of the problem PKICO (S4, S5 and S6). Four sets of algorithms were considered: the direct application of the two hyperheuristic configurations (H), the set of metaheuristics obtained from hyperheuristics (M-H) with fitness computation as FitSP1E, the application of the best metaheuristics not selected automatically (B-M), and the set of the four pure metaheuristics (P-M).

statistically the results presented in Figure 1 and indicates that the metaheuristics obtained with the hyperheuristics (M-H) improve the best results obtained with the best metaheuristic (B-M) reached in the previous work.

# 5 Conclusions and Future Work

Previous fitness results in the determination of kinetic constants of a chemical reaction in heterogeneous phase have been improved by using hyperheuristics based on parameterized schemes for metaheuristics. Specifically, hyperheuristics significantly improve the results of applying pure metaheuristics and also improve the fitness resulting from the application of the best known hybrid metaheuristic. This leads us to think that the hyperheuristic is a good automatic optimization tool that allows us to select optimal metaheuristics.

Similar results were obtained regarding fitness with both hyperheuristics tested. However, it would be advisable to use the reduced hyperheuristic because it significantly reduces the execution time.

Future work could include the application of the same methodology to other optimization problems of chemical parameters and processes. For problems with a high computational cost it is convenient to develop unified parameterized schemes in parallel systems of different architectures: shared memory, message passing, GPU and hybrid-heterogeneous clusters.

## 5.1 Acknowledgements

This work was supported by the Spanish MINECO, as well as European Commission FEDER funds, under grant TIN2012-38341-C04-03.

# References

[1] F. Almeida, D. Giménez, J.-J. López-Espín, and M. Pérez-Pérez. Parameterised schemes of metaheuristics: basic ideas and applications with Genetic algorithms, Scatter Search and GRASP. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 43(3):570–586, 2013.

[2] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. Woodward. *A Classification of Hyper-heuristic Approaches*, pages 449–468. Springer, 2010. In Michel Gendreau, Jean-Yves Potvin, editors, Handbook of Meta-heuristics.

[3] J.-M. Cutillas-Lozano and D. Giménez. Determination of the kinetic constants of a chemical reaction in heterogeneous phase using parameterized metaheuristics. In *ICCS*, 2013.

[4] L.-G. Cutillas-Lozano, J.-M. Cutillas-Lozano, and D. Giménez. Modeling shared-memory metaheuristic schemes for electricity consumption. In *Distributed Computing and Artificial Intelligence - 9th International Conference*, pages 33–40, 2012.

[5] S. S. Dhumal and R. K. Saha. Application of genetic algorithm for evaluation of kinetic parameters of coal pyrolysis. *Journal of Energy & Environment*, 5:112–124, 2006.

[6] L. Elliott, D. B. Ingham, A. G. Kyne, N. S. Mera, M. Pourkashanian, and C. W. Wilson. Genetic algorithms for optimisation of chemical kinetics reaction mechanisms. *Progress in Energy and Combustion Science*, 30(3):297–328, 2004.

[7] H. S. Fogler. *Elements of Chemical Engineering Reaction Engineering*. Prentice Hall, forth edition, 1999.

[8] K. V. Kumar. Linear and non-linear regression analysis for the sorption kinetics of methylene blue onto activated carbon. *Journal of Hazardous Materials*, 137(3):1538–1544, 2006.

[9] T. Lohmann, H. G. Bock, and J. P. Schloeder. Numerical methods for parameter estimation and optimal experiment design in chemical reaction systems. *Industrial and Engineering Chemistry Research*, 31(1):54–57, 1992.

[10] E. Özcan, B. Bilgin, and E. Korkmaz. A comprehensive analysis of hyper-heuristics. *Intelligent Data Analysis*, 12(1):3–23, 2008.

[11] L. N. Plummer, T. M. L. Wigley, and D. L. Parkhurst. The kinetics of calcite dissolution in $CO_2$–water systems at 5 to 60$^o$C and 0.0 to 1.0 atm $CO_2$. *American Journal of Science*, 278:179–216, 1978.